Advanced Hierarchical MPT Modeling with TreeBUGS

Daniel W. Heck

Philipps Universität Marburg

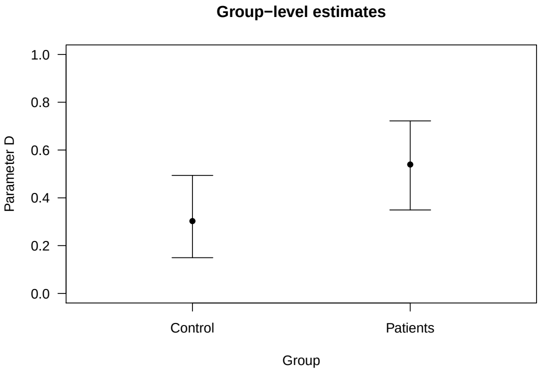
25.02.2020

Advanced Hierarchical MPT Modeling with TreeBUGS

1) Advanced modeling options
   - Between-subject comparisons
   - Within-subject comparisons
   - Linking covariates to MPT parameters
2) Sensitivity/robustness analysis
   - Priors
   - Predictive distributions
   - Simulation

Between-Subject Comparisons

- Often, we are interested in parameter comparisons across groups
- Example: Does the memory parameter D differ for healthy controls vs. schizophrenics?
- Test: Does the group-level parameter μ_D differ across groups?



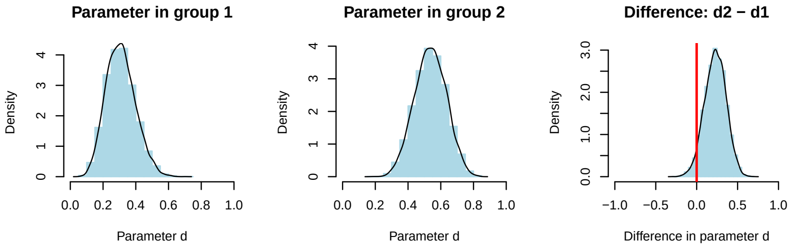
Another Advantage of MCMC: Transformed Parameters

MCMC estimation of transformed parameters

- Based on MCMC samples, we can directly estimate functions (e.g., differences) of parameters

Computational steps

1. Draw MCMC samples
2. Compute transformed parameters for all samples: $\delta^{(t)} = \theta_1^{(t)} - \theta_2^{(t)}$
3. Summarize the distribution of computed values



Between-Subject Comparisons

Parameter comparisons in between-subject designs

1. Fit MPT model to each condition separately
   - Thereby, we assume separate parameters μ and Σ per group
2. Compute differences of group-level parameters for the two fitted models

```
fit1 <- traitMPT(htm_d, "2htm.csv")
fit2 <- traitMPT(htm_d, "2htm_group2.csv")
```

```
diff_between <- betweenSubjectMPT(
  fit1, fit2,              # fitted MPT models
  par1 = "d",              # parameter to test
  stat = c("x-y","x>y"))  # transformed parameters)
diff_between
```

```
##              Mean    SD  2.5%   50% 97.5% Time-series SE n.eff  Rhat R_95%
## d.m1-d.m2 0.174 0.076 0.017 0.175 0.318          0.005   272 1.021 1.073
## d.m1>d.m2 0.984 0.127 1.000 1.000 1.000          0.004   845 1.016 1.021
```

Within-Subject Comparisons

Parameter comparisons in within-subject designs

- Conceptually similar to parameter comparisons in between-subjects designs
- But slightly different estimation

Computational steps

1. Data: Add separate columns for different within-subject conditions
2. Model: Write EQN file for within-subject design
3. MCMC sampling (as usual)
4. Comparison: Compute differences of parameters (transformed parameters)

(1) Data structure for within-subject design

```r
freq_within <- read.csv("2htm_within.csv")
head(freq_within, 3)
```

```
##   high_cr high_fa high_hit high_miss low_cr low_fa low_hit low_miss
## 1      33      17       33        17     40     10      33       17
## 2      44       6       41         9     37     13      34       16
## 3      50       0       50         0     40     10      41        9
```

(2) Model: Function for writing within-subject EQN files

- TreeBUGS provides a function to extend an MPT model to multiple within-subject conditions
- Essentially, model equations are copied and each parameter gets a new label (e.g., `d_condition1` & `d_condition2`)

```
# create EQN file for within-subject manipulations
withinSubjectEQN(htm_d,
                 labels = c("high","low"),  # factor labels
                 constant=c("g"))           # constant parameters
```

```
##          Tree  Category         Equation
## 1  high_target  high_hit           d_high
## 2  high_target  high_hit      (1-d_high)*g
## 3  high_target high_miss (1-d_high)*(1-g)
## 4    high_lure   high_cr           d_high
## 5    high_lure   high_fa      (1-d_high)*g
## 6    high_lure   high_cr (1-d_high)*(1-g)
## 7   low_target   low_hit            d_low
## 8   low_target   low_hit       (1-d_low)*g
## 9   low_target  low_miss  (1-d_low)*(1-g)
## 10    low_lure    low_cr            d_low
## 11    low_lure    low_fa       (1-d_low)*g
## 12    low_lure    low_cr  (1-d_low)*(1-g)
```

(4) Transformed parameters

- The interest is in the difference of a parameter across conditions
- Example: Difference in memory strength $\Delta_d = d_{\text{high}} - d_{\text{low}}$
- Again, we can simply compute any function of interest using transformed parameters
- We get a new set of posterior samples that can be summarized as usual

```
# fit to all conditions:
fit_within <- traitMPT("2htm_within.eqn", "2htm_within.csv")
```

```
# compute difference in d:
diff_d <- transformedParameters(
  fit_within,
  transformedParameters = list("diff_d = d_high - d_low"),
  level = "group")
summary(diff_d)$statistics
```

```
##          Mean            SD      Naive SE Time-series SE
##  0.2828832814  0.0369057519  0.0003551258   0.0012850055
```

Linking Covariates to MPT Parameters

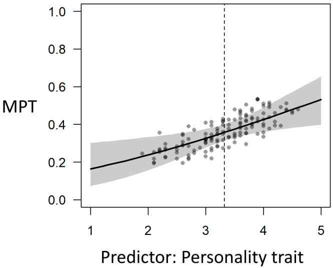
Cognitive Psychometrics (Riefer et al., 2002)

- Using cognitive (MPT) models to learn about interindividual differences
- Linking MPT model parameters to continuous covariates

Interindividual differences

- Idea: Using personality traits or cognitive abilities as predictors for MPT parameters
- Statistical approach in latent-trait MPT: Similar to logistic regression

$$\theta_i = \Phi(\mu + \boxed{\beta \cdot x_i} + \delta_i)$$



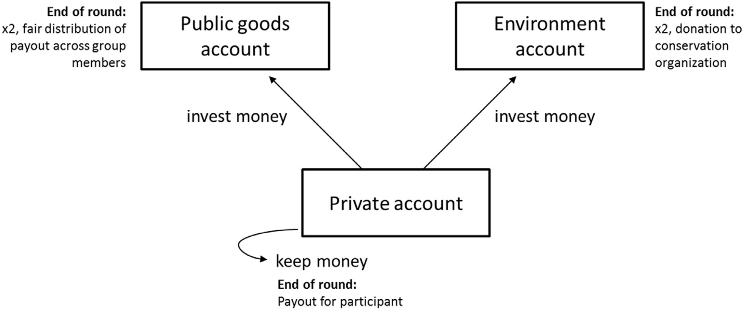
Application: Environmental Psychology

Example: Linking personality to MPT parameters

- "Which is the greater good? A social dilemma paradigm disentangling environmentalism and cooperation"
  - Klein, Hilbig, & Heck (2017). *Journal of Environmental Psychology*)
- Research question: How can we distinguish between 3 types of behavior?
  - Pro-environmental behavior
  - Pro-social behavior
  - Selfish behavior

Application: The Greater Good Game

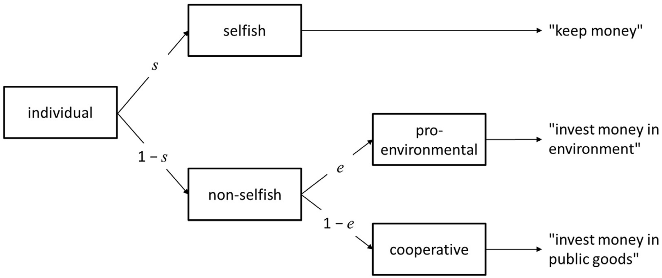
- Greater Good Game
  - Participants decide whether to keep the money for themselves or contribute it to either a public goods or an environment account.
  - Important: Participants are forced to decide between the group and the environment!
  - The game is a variant of a nested public goods game



Application: MPT Model

MPT model for the Greater Good Game

- s = probability of selfish behavior
- e = probability of pro-environmental behavior



Results

- Honesty Humility (= sincerity, fairness) is associated with less selfish behavior
- Selfish behavior decreases from 33.4% to 13.9% for participants $-1/+1$ SD on Honesty Humility

Covariates: Regression

Regression of MPT parameters on covariates

- Example: Predict memory performance d as a function of age
- Statistically, this requires an regression extension to the model
- The latent probit values θ'_i are predicted by a design matrix X:

$$\theta'_i = \mu + X_i\beta + \delta_i$$

Implementation in TreeBUGS

- Requires only two new arguments to provide the data (age of persons) and the regression structure (predict parameter D_n by age)

```
fit_regression <- traitMPT(htm_d, data = "2htm.csv",
                           covData = "covariates.csv",
                           predStructure = list("d ; continuous"))
```

```
round(fit_regression$summary$group$slope[,-6], 2)
```

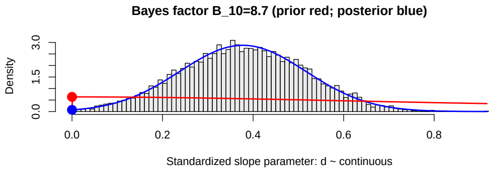
```
##   Mean    SD  2.5%   50% 97.5%  n.eff  Rhat R_95%
##   0.17  0.06  0.05  0.17  0.29 322.00  1.00  1.01
```

Bayes Factor for Covariate

Compute a Bayes factor

- H0: Slope parameter $\beta = 0$
- H1: Slope parameter $\beta \sim$ Cauchy(0, r) (with scale parameter r)
- Method: Savage-Dickey density ratio (Wagenmakers, 2010)
  - Bayes factor H1 vs. H0: prior devided by posterior density (at $\beta = 0$)
  - Only works for simple regression with 1 predictor (Heck 2019)

```
BayesFactorSlope(fit_regression,
                 parameter = "slope_d_continuous",
                 direction = ">",      # H1: positive slope parameter
                 plot = TRUE)          # plot Savage-Dickey density ratio
```




```
##                           BF_0>     BF_>0
## slope_d_continuous 0.1149427 8.699984
```

Covariates: Correlations

Correlation of MPT parameters with external covariates

- In a regression, the inclusion of a covariate changes the estimates of the MPT parameters
- If this is not desirable, one may compute a correlation instead
- New argument `covData`: A data frame or file name with values of the covariate(s)
- TreeBUGS computes the correlation of these covariates with the latent person parameters θ (probit values)

```
fit_cor <- traitMPT(htm_d, data = "2htm.csv",
                    covData = "covariates.csv")   # data with covariate(s)
```

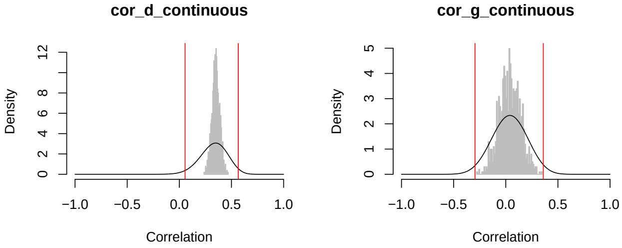
```
# uncertainty with respect to the MPT parameter estimates:
round(fit_cor$summary$group$cor[,-6], 2)
```

```
##                  Mean   SD  2.5%  50% 97.5% n.eff Rhat R_95%
## cor_d_continuous 0.35 0.04  0.28 0.35  0.42  3596    1  1.02
## cor_g_continuous 0.04 0.10 -0.15 0.04  0.24  2008    1  1.00
```

Covariates: Correlations

- Note that the posterior samples of the (descriptive) correlations only reflect uncertainty with respect to the MPT parameters
- We also need to consider the number of participants (sample size)!
- Solution: Use an analytical solution or the posterior distribution of the correlation (Ly et al., 2018)

```
correlationPosterior(fit_cor)
```




```
##                       2.5%   50% 97.5%
## cor_d_continuous   0.055 0.330 0.565
## cor_g_continuous -0.295 0.035 0.360
```

Advanced Modeling: Between-Subject Comparisons via Discrete Predictors

Between-subject designs: Assumptions about the covariance matrix Σ

A) Separate covariance matrix per condition: $\mathbf{\Sigma}_1, \mathbf{\Sigma}_2, \dots$
  - See previous slides: `betweenSubjectMPT(fit1, fit2)`

B) Identical covariance matrix $\mathbf{\Sigma}$ across conditions
  - Similar to ANOVA: "pooled variance" (Rouder & Morey; 2012)
  - Manipulation only affects the mean parameters $\boldsymbol{\mu}$

```
# fit all between-conditions jointly:
fit_between <- traitMPT(
                  htm_d, "2htm.csv",
                  covData = "covariates.csv",
                  predStructure = list("d ; discrete"),  # discrete predictor
                  predType = c("c","f")) # "c" =continuous; "f"=fixed-effects
```

```
# get estimates for the group-specific MPT parameters
gmeans <- getGroupMeans(fit_between)
round(gmeans, 2)
```

```
##                      Mean   SD 2.5%  50% 97.5% p(one-sided vs. overall)
## d_discrete[group_a] 0.52 0.08 0.36 0.52  0.67                     0.02
## d_discrete[group_b] 0.74 0.06 0.60 0.74  0.85                     0.02
```

Combining fixed-effects and random-effects

- In hierarchical MPT models, all parameters are assumed to differ across persons
- Alternative: assume that some parameters are identical for all persons (fixed effects)
- Can be added in TreeBUGS via `restrictions`

```
fit_FE <- traitMPT("2htm.eqn", data = "2htm.csv",
                   restrictions = list("dn=do",  # equality constraint
                                       "g=FE"))  # "FE" = fixed effects
summary(fit_FE)
```

Sensitivity/Robustness Analysis

Changing Priors

Define different priors

- Prior distributions in the latent-trait MPT necessary for:
  - Latent (probit-) mean μ
  - Latent (probit-) covariance matrix Σ: scaled inverse Wishart with
    - Prior matrix V
    - Degrees of freedom `df`
    - Scaling parameter ξ
- Example: We assume that guessing probabilities are around 50%

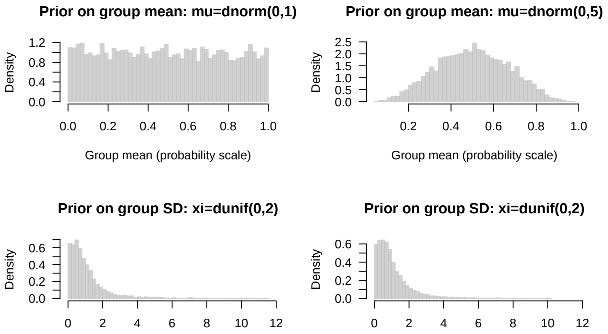
```
fit <- traitMPT(eqnfile="htm.txt", data="responses.csv",
                restrictions=list("dn=do"),

                mu = c(dn = "dnorm(0,1)",   # default prior
                       g = "dnorm(0,5)"),   # prior focused around 50% guessing
                xi = "dunif(0,2)",          # less disperson of MPT parameters
                V = diag(2),                # default
                df = 2 + 1)                 # default
```

Understanding Priors

What do the priors actually mean?

- Draw samples from the prior
- Plot mean/SD of MPT parameters

```
plotPrior(prior =
            list(mu = c(dn = "dnorm(0,1)",  # default prior
                        g =  "dnorm(0,5)"), # prior focused around 50%
                 xi="dunif(0,2)",           # smaller scale for group SD
                 V= diag(2), df = 3))       # default Wishart prior
```

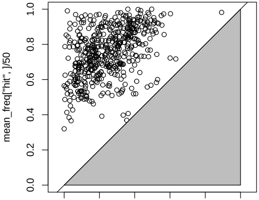


Prior Predictive Sampling

Prior predictive distribution

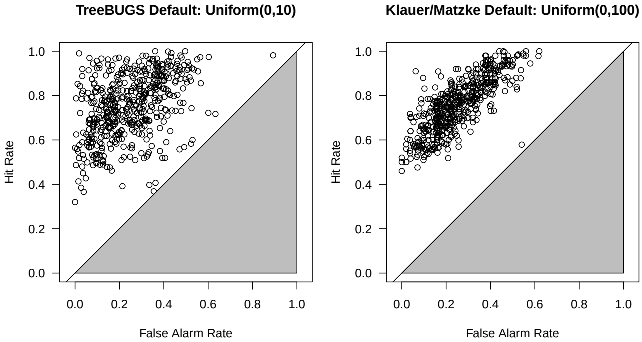
1. Draw samples from the prior
2. Draw new data (response frequencies)
3. Assess predicted data (e.g., plots or descriptive statistics)

```
pp <- priorPredictive(prior = list(mu = "dnorm(0,1)", xi="dunif(0,10)",
                                   V=diag(2), df=2+1),
                      eqnfile = htm, restrictions = list("dn=do"),
                      numItems = c(old = 50, new = 50),
                      N = 50,  M = 500)     # number of participants/samples

# compute and plot predicted values for the average hit/FA rates (in ROC space)
mean_freq <- sapply(pp, colMeans)
par(mar=c(4,5,.1, .1))
plot(mean_freq["fa",]/50, mean_freq["hit",]/50, asp = 1, xlim =0:1, ylim=0:1)
polygon(c(0,1,1), c(0,0,1), col = "gray")
abline(0, 1)
```



Excursion: Different default priors for the scale parameter ξ

1. TreeBUGS (Heck et al., 2018): $\xi \sim$ Uniform$(0, 10)$
2. Klauer (2010) and Matzke et al. (2015): $\xi \sim$ Uniform$(0, 100)$

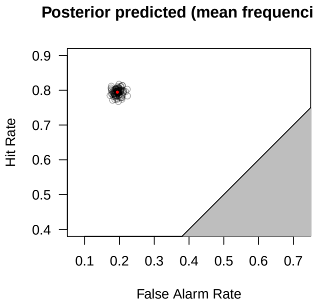


Posterior Predictive

Posterior Predictive Distribution

- What data does the fitted model predict?
- Use posterior samples of the *parameters* to draw new samples of the *data* (i.e., predicted response frequencies)
  - Note: These are the basis of posterior-predictive checks (T1 and T2 statistics)

```
postpred <- posteriorPredictive(fit, M = 100, nCPU = 4)
```



Simulations

Sensitivity and robustness analysis

- Goals:
  - Assessing the impact of specific priors
  - Estimating the necessary sample size for specific analysis
- Basic steps of a simulation:
  1. Generate data from a specific model
  2. Fit (correct or wrong) model with specific priors
  3. Replicate multiple times using a `for`-loop
  4. Summarize results (e.g., parameter estimates)

```r
# standard, fixed-effects MPT (generate data for one person)
sim <- genMPT(theta = c(d = .7, g = .5),
              numItems = c(target = 50, lure = 50),
              eqnfile = htm_d)

# hierarchical MPT (generate a complete table of frequencies)
sim2 <- genTraitMPT(N = 100, eqnfile = htm_d,
                    mean = c(d = .7, g = .5),
                    sigma = c(d = .4, g = .2),
                    rho = diag(2))
```

References

Heck, Daniel W. 2019. "A Caveat on the Savage-Dickey Density Ratio: The Case of Computing Bayes Factors for Regression Parameters." *British Journal of Mathematical and Statistical Psychology* 72: 316–33. https://doi.org/10.1111/bmsp.12150.